








TÉCNICO LISBOA

Relatório de Sistemas Distribuídos

Grupo: T_18_24

14-05-2014

		
Adílio Santos Nº 70522	David Limpo Nº 70630	Francisco Pedreira Nº 71033
		
João Brás Nº 70598	José Andrade Nº 70451	Rui Santos Nº 71042

Índice

Registo Fatura	3
Replicação Passiva	3
Trabalho Futuro	4
Cheque Refeição	5
Segurança.....	5
Trabalho Futuro	6

Registo Fatura

Replicação Passiva

Foi-nos proposto que realizássemos um projecto em que o objectivo era garantir a tolerância a faltas, utilizando replicação passiva.

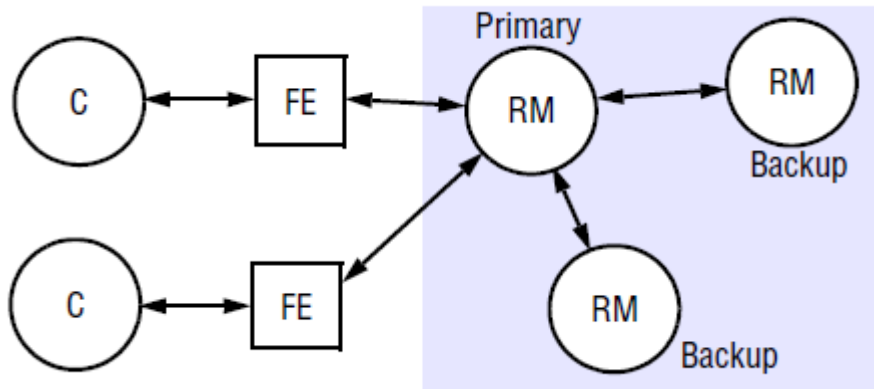


Figura 1: Arquitectura da replicação passiva.

A replicação passiva consiste na interacção do cliente com um servidor principal enquanto os restantes servidores estão de reserva. Quando os servidores de reserva detectam que o servidor primário falhou, um deles torna-se no primário, assumindo as suas funções. Sempre que o servidor primário executa um serviço, replica-o para os secundários de modo a que todos os servidores se mantenham sincronizados.

Baseando-nos neste modelo de replicação, foi implementado um servidor primário capaz de interagir com N servidores secundários (servidores de backup).

Por forma a garantir a persistência dos dados, antes de os servidores serem lançados, as suas bases de dados deverão ser devidamente inicializadas.

Os servidores secundários necessitam de ser lançados antes do primário, de maneira a gerar os stubs que o servidor primário precisará de usar para comunicar com os secundários.

De seguida, é lançado o servidor primário que executa uma thread que obtém todos os endereços URL dos servidores secundários registados no UDDI.

Esta lista de endereços é posteriormente enviada a todos os secundários.

Isto é feito para que quando o servidor primário falhe, o servidor correspondente ao primeiro endereço da lista assuma o controlo.

Este procedimento envolve a substituição do antigo servidor primário no UDDI pelo novo.

É necessário que os secundários saibam quando assumir o controlo.

Para tal, o servidor primário precisa de indicar aos restantes que está activo enviando mensagens “I’m alive” a cada P unidades de tempo (considerou-se que P seria 7 segundos).

Se as mensagens deixarem de ser enviadas, assume-se que o primário está inactivo. Como tal, um dos secundários assume as suas funções.

De modo a garantir consistência sequencial, quando é invocado um serviço no servidor principal pelo cliente, percorre-se a lista de secundários invocando o mesmo serviço em cada um deles.

Desta forma, garante-se que todas as alterações efectuadas no servidor principal se repercutem para todos os servidores secundários mantendo-se a consistência entre eles.

Todo este processo é feito automaticamente, sem que o cliente se aperceba que houve uma falha em um ou mais servidores.

Trabalho futuro

Tornar o servidor primário independente dos servidores secundários.

Neste momento, é necessário correr um servidor secundário para tornar o serviço “I’m Alive” disponível por forma a que o servidor primário possa aceder ao contrato dinamicamente para gerar os *stubs* necessários.

A solução passaria por guardar o contrato WSDL localmente no servidor primário, fazer o *import* dos *stubs* do serviço “I’m Alive” e, desta forma, o servidor primário tornar-se-ia independente dos serviços oferecidos pelo servidor secundário.

Cheque Refeição

Segurança

A segunda entrega do Projecto de SD do Cheque Refeição consiste na criação de mecanismos de segurança que permitam que vários clientes se possam ligar a um servidor de forma segura, autêntica e confidencial. Como a criação e uso de cheques é uma operação crítica para o cliente e a informação dos cheques é demasiado importante para poder ser acedida por pessoas alheias o grupo realizou os seguintes mecanismos para prevenir fugas de informação e tentativas de acesso a cheques indevidas:

1. Criação e geração de chaves

Tanto o servidor como o cliente contem um conjunto de chaves, pública e privada, criadas de forma assimétrica. Para isso foi criado uma classe chamada **AsymCrypto** cujo construtor cria um *Secure Random Number* que será usado para a geração das chaves públicas e privadas. Essa geração é feita pelo comando:

"nome do Objecto AsymCrypto".generateKeys();

Desta forma é possível aceder ao objecto e efectuar um `getPubKey`, para obter a chave pública e um `getPrivKey`, para obter a chave privada.

2. Criação de uma Assinatura para validar a autenticidade do Cliente.

O Cliente possui de uma classe **Signature**, que permite a geração de uma assinatura para que o servidor saiba sempre quem é a pessoa a aceder ao servidor. A assinatura é gerada com a chave privada do cliente e o servidor posteriormente verifica a autenticidade do mesmo com a chave pública do cliente.

3. Criação de Ids seguros para os cheques.

Os ids dos cheques são criados usando a função que permite gerar chaves públicas e privadas de forma a garantir a maior segurança possível. O id público corresponde à chave pública e o segredo corresponde à chave privada. Nas operações de endossar, quando é criado o 2 segredo, é gerada uma nova chave privada que é atribuída como Segredo2.

4. Operação listar

Foram tomadas as medidas necessárias para que quando um Cliente utiliza a operação listar apenas o id público do cheque é visível, deste modo garante que mesmo que se consiga listar os cheques de um determinado cliente não é apresentada a informação necessária para o seu acesso.

Trabalho futuro

Para melhorar a segurança poderia ter sido usado algoritmos mais complexos na geração das chaves públicas e privadas. O servidor poderia também guardar na base de dados todas as chaves públicas.

Neste momento, não é possível garantir que se um cliente endossa, de forma offline o cheque a um beneficiário, que esse seja capaz de aceder ao cheque. A solução seria elaborar um método mais avançado que permitisse que um beneficiário consiga aceder ao cheque do cliente sabendo o id publico + segredo do cliente que lhe passou o cheque. No entanto esta operação faria com o sistema não fosse plenamente seguro pois poderia ser alvo de tentativas de falsificação de identidade, dando ao cliente toda a responsabilidade por endossar o cheque.